L #5

CJNE (compare and jump if not equal)

CJNE A, #data, label

CJNE A, direct, label → memory location

CJNE Rn, #data, label

CJNE @Ri, #data, label

→ R3 - 25h
- =0 → R3 = 25h → cy = 0
- =+ → R3 > 25h → cy = 0
- =− → R3 < 25h → cy = 1

CJNE R3, #25h, NEXT

R3 = 25h

SJMP Exit

R3 ≠ 25h

NEXT: JC THERE

JNC

R3 > 25h

SJMP Exit

THERE:

R3 < 25h

Exit:

write a program to check the content of R_3 as follows

if $R_3 = 25h$, then save 55h in A

if $R_3 > 25h$, then " AAh in A

if $R_3 < 25h$, " " 'L' in A

```
        CJNE   R3, #25h, NEXT
        MOV    A, #55h
        SJMP   Exit
NEXT:   JC     Less
        MOV    A, #0AAh
        SJMP   Exit
Less:   MOV    A, #'L'
Exit:
```

Addressing modes:

- Addressing mode is a method of specifying operands
- " modes are various ways of accessing data.

* Data could be:

① Memory → RAM / ROM

② Register

③ immediate (in the instruction itself).

---

* 8051 has 5 addressing modes

1- Immediate addressing mode ←

2- Register " " ←

3- Direct " "

4- Register indirect " "

5- Indexed addressing mode

} memory

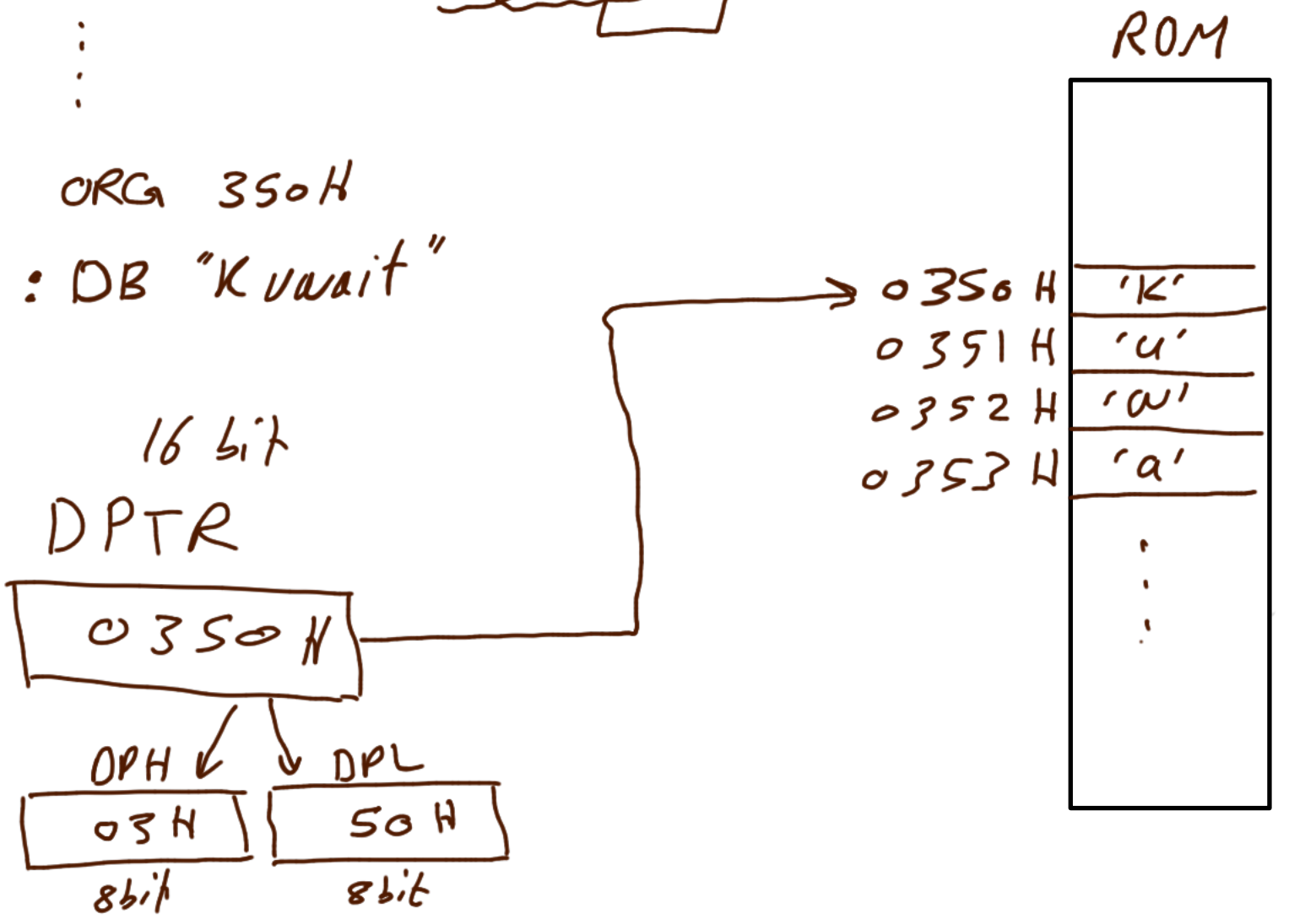
- What are the addressing modes for both source and destination ? (4)

(1) Immediate addressing mode (#)

```
MOV   R4, #28h
            └→ immediate
```

```
NUM1  EQU  50
:
MOV   A, #NUM1      ; A = 50 → = 2.H
            └→ immediate
```

```
MOV   DPTR, #mydata
:
:
ORG   350H
mydata : DB "Kuwait"
```



MOV DPTR, #0350H

MOV DPL, #50H ←

MOV DPH, #03H → immediate

Register ←

DPTR

0350H

---

② Register addressing mode

MOV A, R4

Register ← → Register

MOV A, DPH

---

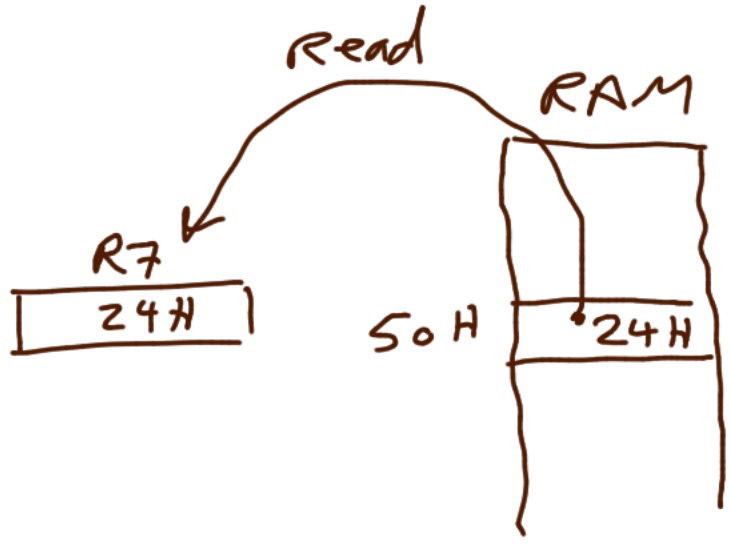
③ Direct addressing mode

(00H → 7FH) RAM

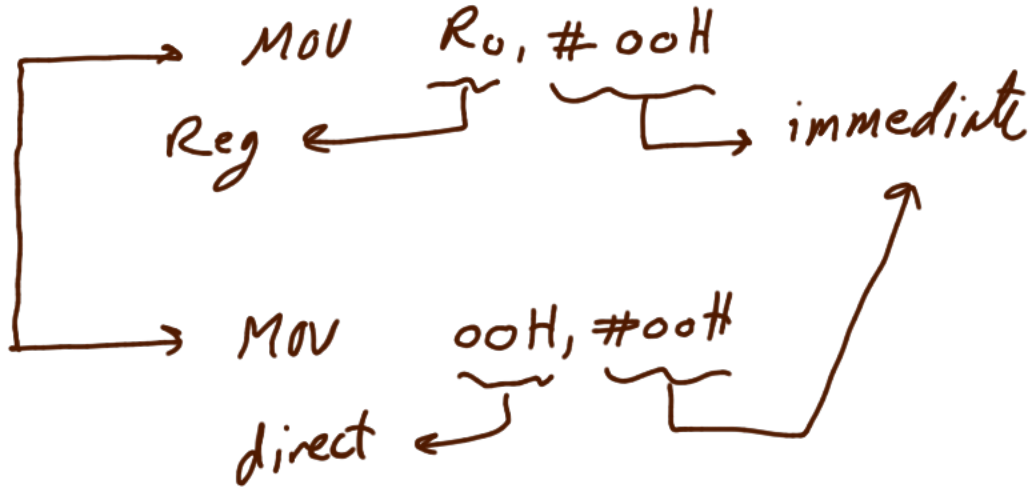
(80H → FFH) SFR

MOV R7, 50H

Register ← → direct

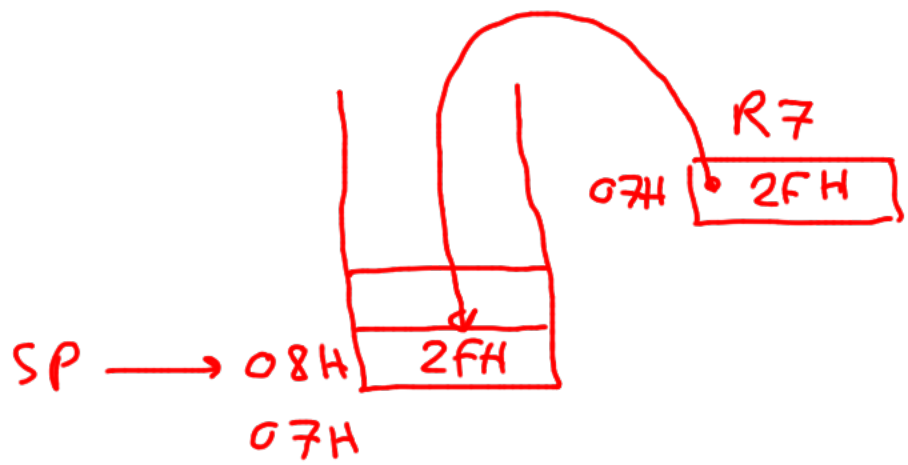


MOV 40H, A → Reg.
direct ←



MOV 40H, 30H ✓

Mov R7, #2FH
Reg
imm
PUSH 7
direct
PUSH 07H



POP 4
direct

push and pop use only direct addressing mode. (7)

---

push the contents of R7 from bank 3 in to stack

PUSH 7 ✗

PUSH 1FH ✓

SETB 2FH → direct

---

(4) Register indirect

R0 and R1 are used as pointers to memory @R0 @R1
locations (00H → 7FH) RAM
or to external memory (00H → FFH)

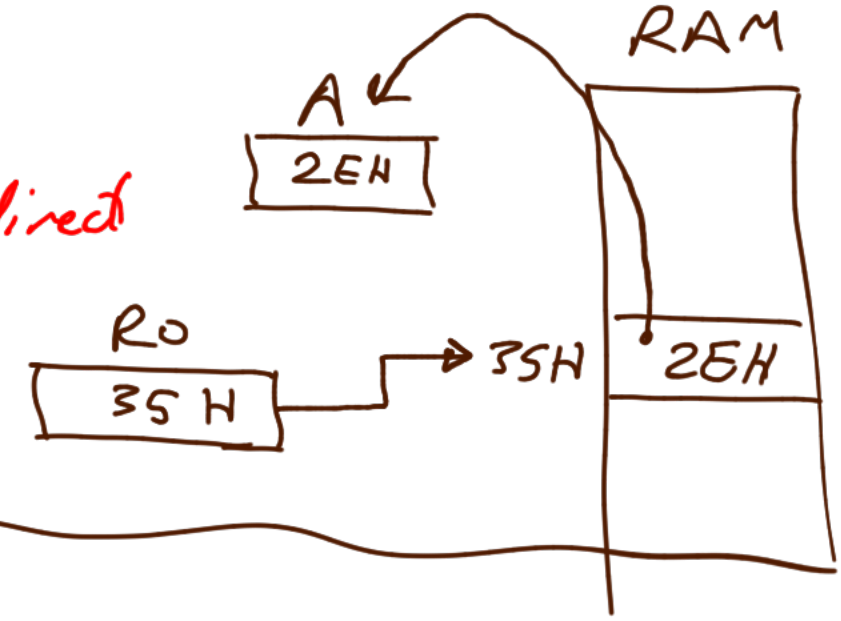
@DPTR

DPTR is used as pointer to external
RAM or ROM (0000H → FFFFH)

MOV R0, # 35H — Reg, imm

MOV 35H, # 2EH — direct

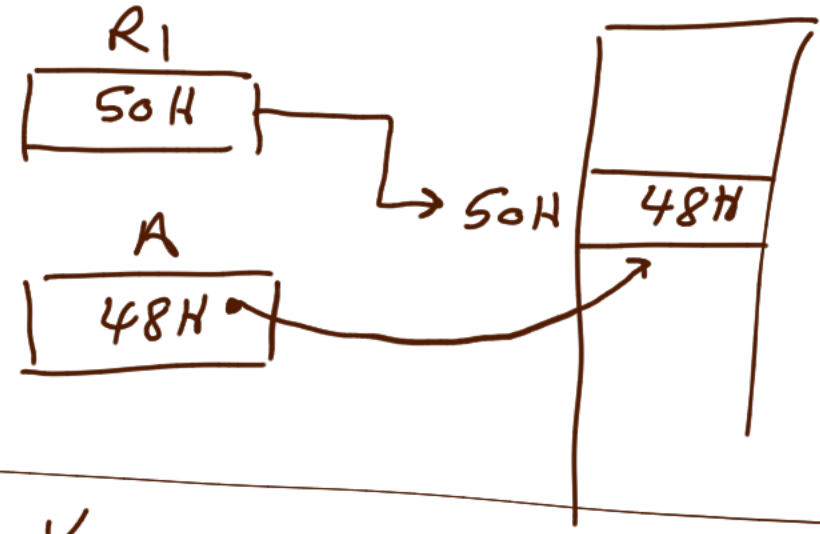
MOV A, @R0 — Reg, Register indirect



MOV R1, # 50H

MOV A, # 48H

MOV @R1, A

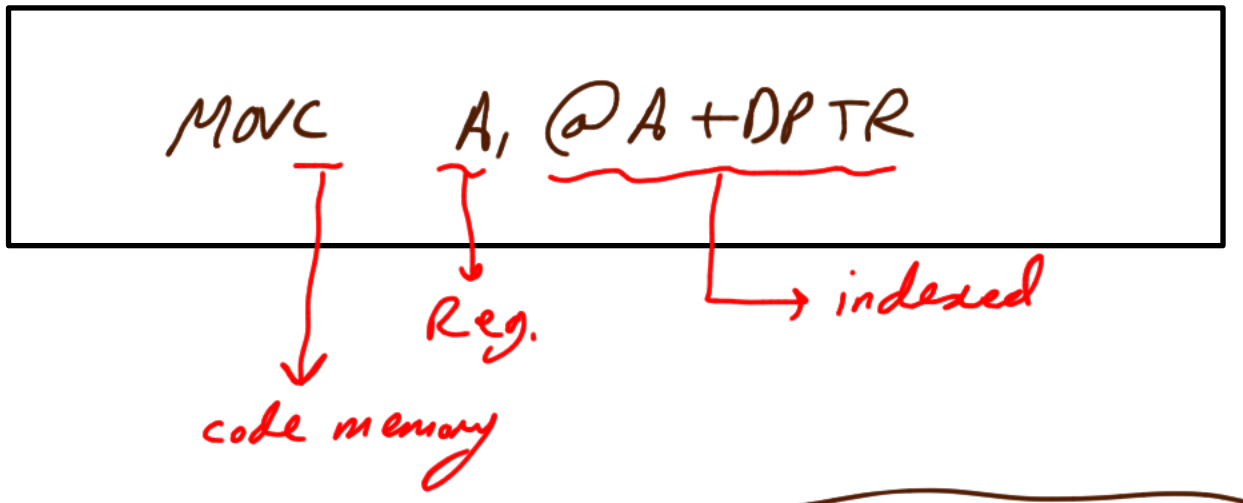


MOV @R0, @R1 X

only R0 and R1 are used as pointers @R3 X

⑤ Indexed

(@Reg + Reg)




```
MOV   A, #3
MOV   DPTR, #0200H
MOVC  A, @A+DPTR
.
.
.
ORG   0200H
DB    12, 24H, 0C8H, 77H, 99H
```